

In the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

1.-13. (Cancelled)

14. (Currently Amended) A machine-readable medium having instructions recorded thereon, such that when the instructions are read and executed by a processor in a first computing system connected to a network, the first computing system performs a method comprising:

receiving, at the first computing system, a request for a web page from a second computing system, the requested web page having content;

creating on the first computing system a page object having references to component objects in response to the received request for information, the page object being created based on a page file, each component object of the page object representing a user control within the page file, wherein creating the page object includes:

retrieving from an output cache any component object that represents one of the user controls of the page file and is contained in the output cache, and

retrieving from another source executable code for any component object that represents one of the user controls of the page file and is not contained in the output cache and instantiating the executable code to create the component object;

determining whether any of the component objects referenced by the page object correspond with a user control that supports output caching; and

caching the component object in the output cache if the component object corresponds with a user control that supports output caching, the component object cached using a cache key that comprises an identifier that is unique for each occurrence of the component object, the cache key being created when the page object is created;

inserting the retrieved component objects of the page object and the created component objects of the page object into a hierarchical tree data model at the first computing system, each component object being linked to a prior component object if a prior component object exists, and each component object being linked to a next component object if a next component object exists;

processing the components of the hierarchical tree data model to create a renderable page at the first computing system; and

sending the created renderable page from the first computing system to the second computing system.

15. (Previously Presented) The medium of claim 14 wherein:

the user control including an output caching directive, wherein caching the component object in the output cache comprises caching the component object according to the output caching directive.

16. (Previously Presented) The medium of claim 14, wherein the contents of the created renderable page comprises an HTML specification for a web page.

17. (Previously Presented) The medium of claim 15, wherein;

the step of processing the created objects comprises processing each one of the components individually.

18. (Previously Presented) The medium of claim 17, further comprising:

creating the hierarchical tree data model including each of the components and a hierarchical relationship among the components, the data model being used during the step of processing the components.

19. (Previously Presented) The medium of claim 15, wherein the output caching directive includes a time duration during which the component object is permitted to reside in the output cache.

20. (Previously Presented) The medium of claim 19, wherein the output caching directive includes an attribute indicating a condition for varying the component object to be stored in the output cache.

21. (Previously Presented) The medium of claim 20, wherein the attribute indicates that the component object is to be stored in the output cache according to a type of browser used by the second computing system.
22. (Previously Presented) The medium of claim 20, wherein the attribute indicates that the component object is to be stored in the output cache according to values of at least one parameter.
23. (Previously Presented) The medium of claim 14, further comprising providing, on the first computing system, performance counters to monitor output caching performance.
24. (Original) The medium of claim 23, wherein the performance counters include:
an output cache hit counter to count a number of requests serviced from the output cache;
and
an output cache miss counter to count a number of failed output cache requests.
25. (Original) The medium of claim 23, wherein the performance counters include an output cache turnover rate to count a number of additions and removals to the output cache per second.
26. (Original) The medium of claim 23, wherein the performance counters include an output cache hit ratio to keep track of a percentage of total requests serviced from the output cache.
27. (Currently Amended) A method for providing a response to a request for information from a client computing system to a server computing system having an output cache for storing static portions of web pages, the method comprising:
receiving a request from the client computing system for a web page having a plurality of components, each of the components of the requested web page being either a static component or a dynamic component;
generating the requested web page including:
determining whether an output cache on the server computing system contains any static components of the web page;

retrieving each of the static components contained in the output cache, wherein each of the static components is cached using a cache key that comprises an identifier that is unique for each occurrence of the static components, the cache key being created when each static component is created;

creating at the server computing system each of the static components not contained in the output cache by retrieving executable code for each respective component from another source and instantiating the retrieved executable code;

determining whether any dynamic components correspond to user controls that support output caching;

creating at the server computing system each of the dynamic components by processing each dynamic component corresponding to a user control including retrieving the executable code from the file and instantiating the retrieved executable code, and by processing each dynamic component that does not correspond to a user control; and

assembling the static components and the dynamic components into a hierarchal data model at the server computing system, the static components and dynamic components being linked to each other in the hierarchical data model;

generating contents for the web page by processing each of the static components and each of the dynamic components of the hierarchal data model; and

sending the generated contents to the client computing system.

28. (Currently Amended) A method comprising:

receiving at a server computing device from a client computing device a request for a web page;

retrieving at the server computing device a page file associated with the requested web page, the page file including control references, which include instructions for obtaining associated page components of the requested web page, each control reference including either a user control or a server control, the instructions of each server control being stored within the page file, the instructions of each user control being stored in a separate file;

examining the page file at the server computing device to identify whether each of the control references is a user control or a server control;

examining each user control at the server computing device to determine whether the user control supports output caching including accessing the separate file and analyzing the instructions of the user control to determine whether the instructions contain an output caching directive;

determining whether the page component associated with each user control that supports output caching is available at a cache of the server computing device;

obtaining from the cache the page component associated with each user control that supports output caching and that is available at the cache of the server computing device, the page component cached using a cache key that comprises an identifier that is unique for each occurrence of the page component, the cache key being created when the page component is created;

generating the page component associated with each user control that supports output caching and is not available at the cache of the server computing device;

generating the page component associated with each user control that that does not support output caching including generating the page component based on the instructions associated with the user control;

generating the page component associated with each server control; and

inserting the generated page components into a hierarchical tree model at the server computing device, each page component being linked to a prior page component if a prior page component exists, and each page component being linked to a next page component if a next page component exists.

29. (Previously Presented) The method of claim 28, wherein the output caching directive indicates a length of time for which the associated page component is to be stored in the cache on the server computing system.

30. (Previously Presented) The method of claim 28, wherein the page file comprises an HTML specification for a web page.

31. (Previously Presented) The method of claim 28, further comprising arranging the page components into a data model to facilitate rendering the requested web page based on the page components.
32. (Previously Presented) The method of claim 31, wherein arranging the page components into the data model comprises arranging the page components into a hierarchical tree data model.
33. (Previously Presented) The method of claim 28, wherein generating the page component associated with each user control that supports output caching and is not available comprises:
retrieving the instructions that are associated with the page components from the respective separate file; and
generating the page components based on the retrieved instructions.
34. (Previously Presented) The method of claim 28, further comprising:
storing in the cache of the server computing device any generated page component that supports output caching and that is not available at the cache of the server computing device.
35. (Previously Presented) The method of claim 34, wherein the output caching directive indicates a condition for varying the page component to be stored in the cache on the server computing device.
36. (Previously Presented) The method of claim 28, wherein the output caching directive indicates the page component is to be stored in the cache according to a type of browser used by the client computing system.
37. (Previously Presented) The method of claim 28, wherein the output caching directive indicates the page component is to be stored in the cache according to values of at least one parameter listed in the page file.
38. (Previously Presented) The method of claim 28, further comprising providing on the server computing system at least one performance counter to monitor caching performance.

39. (Previously Presented) The method of claim 38, wherein the performance counter includes:

- an output cache hit counter to count a number of requests serviced from the cache;
- an output cache miss counter to count a number of failed cache requests; and
- an output cache turnover rate to count a number of additions and removals to the cache per second.

40. (Previously Presented) The method of claim 38, wherein the performance counter includes an output cache hit ratio to keep track of a percentage of total requests serviced from the cache of the server computing device.